In the Claims:

1. (Original) A parallel processing array comprising a plurality of processing elements (PEs), each processing element receiving a common instruction and comprising: a multiplexer for receiving said common instruction; an arithmetic logic unit, connected to said multiplexer, for processing the received instruction in association with an accumulator and a flag register; characterized in that one or more of the processing elements in the processing array further comprises a storage element having at least one storage location, the storage element configured to be indirectly addressable by the received instruction, thereby enabling the processing of data dependent operations to be performed.

2. (Original) A parallel processing array as claimed in claim 1, wherein the storage element comprises: an input data port for receiving data to be stored; an index signal for addressing a storage location in the storage element; and an output port for outputting data from the storage element.

3. (Original) A parallel processing array as claimed in claim 2, wherein the input data port of the storage element is connected to receive data from an input multiplexer, the input multiplexer being configured to pass accumulator data or coefficient data.

4. (Previously presented) A parallel processing array as claimed in claim 2, wherein the index signal is received from an index multiplexer, the index multiplexer being configured to selectively pass accumulator data or coefficient data, or part of the received instruction.

5. (Previously presented) A parallel processor as claimed in claim 3, wherein the input multiplexer and/or index multiplexer is controlled by the received instruction.

6. (Previously presented) A parallel processing array as claimed in claim 1, wherein the storage element is configured to provide the processing element with a coefficient

based on the data to be processed.

7. (Previously presented) A parallel processing array as claimed in claim 6, wherein the input multiplexer is configured to pass accumulator data to the storage element when coefficient data is being stored in a storage location defined by the index signal.

8. (Original) A parallel processing array as claimed in claim 6, wherein the input multiplexer is configured to pass coefficient data to the storage element, stored in a storage location defined by the index signal.

9. (Previously presented) A parallel processing array as claimed in claim 7, wherein the index signal is defined by coefficient data received by the index multiplexer.

10. (Previously presented) A parallel processing array as claimed in claim 7, wherein the index signal is defined by accumulator data received by the index multiplexer.

11. (Previously presented) A parallel processing array as claimed in claim 1, wherein the storage element is configured to provide a local look-up table for the processing element.

12. (Original) A parallel processing array as claimed in claim 11, wherein the input multiplexer is configured to pass coefficient data to the storage element for storage in a location defined by the index signal.

13. (Original) A parallel processing array as claimed in claim 12, wherein the index signal is defined by accumulator data received by the index multiplexer.

14. (Original) A parallel processing array as claimed in claim 11, wherein the input multiplexer is configured to pass a first part of the coefficient data as the data to be stored in the storage element, and the index multiplexer arranged to pass the other part of the coefficient data as the index signal defining the storage address.

15. (Previously presented) A parallel processing array as claimed in claim 1, further comprising a register for storing data between the output of the storage element and the input of the multiplexer.

16. (Previously presented) A parallel processing array as claimed in claim 1, wherein the processing array is a single instruction multiple data (SIMD) processing array.

17. (Original)A method of processing data in a parallel processing array comprising a plurality of processing elements (PEs), each processing element receiving a common instruction and comprising a multiplexer for receiving said common instruction, and an arithmetic logic unit, connected to said multiplexer, for processing the received instruction in association with an accumulator and a flag register, the method comprising the steps of: providing a storage element in one or more of the processing elements in the processing array, the storage element having at least one storage location; configuring the storage element to be indirectly addressable by the received instruction; and processing data dependent operations using the storage element.

18. (Original) A method as claimed in claim 17, further comprising the steps of: providing an input data port in the storage element for receiving data to be stored; providing an index signal for addressing a storage location in the storage element; and providing an output port for outputting data from the storage element.

19. (Original) A method as claimed in claim 18, further comprising the steps of connecting the input data port of the storage element to receive data from an input multiplexer, and configuring the input multiplexer to pass accumulator data or coefficient data.

20. (Previously presented) A method as claimed in claim 18, further comprising the step of providing an index multiplexer for providing the index signal, and configuring the index multiplexer to selectively pass accumulator data or coefficient data, or part of the

received instruction.

21. (Previously presented) A method as claimed in claim 19, further comprising the step of controlling the input multiplexer and/or index multiplexer with the received instruction.

22. (Previously presented) A method as claimed in claim 17, further comprising the step of configuring the storage element to provide the processing element with a coefficient based on the data to be processed.

23. (Previously presented) A method as claimed in claim 22, further comprising the step of configuring the input multiplexer to pass accumulator data to the storage element when coefficient data is being stored in a storage location defined by the index signal.

24. (Original) A method as claimed in claim 22, further comprising the step of configuring the input multiplexer to pass coefficient data to the storage element, and storing the coefficient data in a storage location defined by the index signal.

25. (Previously presented) A method as claimed in claim 23, wherein the index signal is defined by coefficient data received by the index multiplexer.

26. (Previously presented) A method as claimed in claim 23, wherein the index signal is defined by accumulator data received by the index multiplexer.

27. (Previously presented) A method as claimed in claim 17, further comprising the step of configuring the storage element to provide a local look-up table for the processing element.

28. (Original) A method as claimed in claim 27, wherein the input multiplexer is configured to pass coefficient data to the storage element for storage in a location defined by the index signal.

29. (Original) A method as claimed in claim 28, wherein the index signal is defined by accumulator data received by the index multiplexer.

30. (Original) A method as claimed in claim 27, further comprising the step of configuring the input multiplexer to pass a first part of the coefficient data as the data to be stored in the storage element, and arranging the index multiplexer to pass the other part of the coefficient data as the index signal defining the storage address.

31. (Previously presented) A method as claimed in claim 17, further comprising the step of providing a register for storing data between the output of the storage element and the input of the multiplexer.

32. (Previously presented) A method as claimed in claim 17, wherein the processing array is a single instruction multiple data (SIMD) processing array.